AD-A 153 049

AD A153049

BRL

MEMORANDUM REPORT BRL-MR-3433

TARGET: A MODEL FOR DETERMINING HITS ON AN IRREGULARLY SHAPED TARGET

Fred L. Bunn
Sue B. Hinman

February 1985

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MEMORANDUM REPORT BRL-MR-3433	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TARGET: A MODEL FOR DETERMINING HITS ON AN IRREGULARLY SHAPED TARGET		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Fred L. Bunn, Sue B. Hinman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: AMXBR-SECAD Aberdeen Proving Ground, MD 21005-5066		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L162618AH80
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: AMXBR-OD-ST Aberdeen Proving Ground, MD 21005-5066		12. REPORT DATE February 1985
		13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Combat
Hit Probability
Target
Polyhedra

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents a program for finding whether a shot will hit an irregularly shaped target. The program uses the projectile velocity and speed and the coordinates of the target's corners to determine whether the target was hit. If desired, the program will tell which faces of the target were entered or exited. The target may be complex as desired. It is a useful program as it stands, but it is designed to be imbedded in stochastic simulations of combat where its utility would be even greater. This report describes the input, output, and mathematics of the algorithm and contains a copy of the Fortran 77 program.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

This report documents a Fortran 77 program for finding whether a projectile hits an irregularly shaped target. It discusses input preparation, tells about the various levels of output provided, and describes the mathematics used by the program. The input and output sections give a sample case which may be used for testing the program. A copy of the program is included in the appendix.

This program determines whether a projectile hits a complex target and if desired, which faces were entered or exited. The target may be composed of one or more polyhedra. As examples, the target could be several tanks, a hollow box, or a donut shaped polyhedron. Restricting the target to polyhedra is not a severe limitation because solids with curved surfaces may be represented to any desired accuracy by increasing the number of polygonal faces. Currently, the model handles targets with up to 20 corners and 20 faces but this is easily increased by changing the three parameter statements in the program.

INPUT DATA

To obtain successful results the input data must be prepared in a specific order. If the order is not followed exactly, the output data will be incorrect. First make sure you have numbered all of the corners and faces of your polygon. The polygons used in this program must be convex polygons. This is not really a limitation as a concave polygon may be broken down into several convex polygons. The coordinate system used in this program is in three dimensions so that the x axis is traveling East, the y axis North, and the z axis Up.

Line 1 should contain the level number. If level=1, the program simply prints whether the target was hit or not. If level=2, it prints an input echo and tells which faces were entered or exited. And, if level=3, it prints the input echo, the results of intermediate calculations, and which faces were entered or exited, if any. DO NOT place a decimal point after the number.

Line 2 should contain the coordinates of the projectile's velocity. The three coordinates (x,y,z) should be placed next to each other horizontally and separated by either commas and/or spaces. The x coordinate should be first, followed by the y, and then the z. DO place a decimal point after the numbers.

Line 3 should contain the coordinates of the projectile. They should be based on the aim point of the projectile to the target. They should be entered exactly like the velocity coordinates in the line above. DO place a decimal point after the numbers.

Line 4 should be the number of corners. The program is set to allow up to 20 corners on the target, so be sure that your number is less than or equal to 20. DO NOT place a decimal point after the number.

Lines 5-n are for the coordinates of the corners. Each corner's coordinates should be entered like those of the projectile's velocity. Corner number one's coordinates will be on the first of these, line number two's on the next, and so forth until you are completed. DO NOT leave any blank lines after these. DO make sure that the number on line four is the same as the number of sets of corner coordinates. DO place a decimal point after the numbers.

Line n+1 should contain the number of faces the target has. The program is set to allow up to 20 faces on the target, so check the number before trying to run the program. DO NOT place a decimal point after the number.

Lines n+2 and following will contain the numbers of the corners for each of the faces, one face per input line. The numbers of the corners should be read in a clockwise direction around the edge of the face starting from any corner on the face. The program allows for six corners per face. If a face has less than six corners just fill in the extra spaces with zeros. Make sure the faces are listed in order as you have numbered them. Face number one should be in the first row, two in the second, and so forth until you have completed them. DO NOT place a decimal point after the numbers. When you have completed your input file you may check it with the sample below to double check your accuracy. The right hand column tells you what is found in each row.

Figure 1 shows a tank turret corresponding to the data in Table 1.

OUTPUT DATA

The program generates output in three levels of detail depending on the first input value. You may select the level you think is most appropriate for your immediate purposes. Level one simply prints whether the target was hit or not. Level two prints an input echo and tells which faces were entered or exited. Level three prints the input echo, the results of intermediate calculations, and which faces were entered or exited, if any.

Level one prints one of the following messages:

```
Yes, it was hit.

No, it was not hit.
```

TABLE 1. SAMPLE INPUT FOR A TANK TURRET

Sample Input Data	
As seen in file	Explanation
1	level of output datail
3., 4., 0.	beginning velocity coordinates
-2., 0., 1.	beginning projectile coordinates
12	number of corners
.9, -4.3, 1.5	coordinates of corners
.9, -5.3, 0.	"
-.9, -4.3, 1.5	"
-.9, -5.3, 0.	"
-3.7,-3.2,1.5	"
-4.8, -3.9, 0.	"
-3.7, 6.2, 1.5	"
-4.8, 7.5, 0.	"
3.7, 6.2, 1.5	"
4.8, 7.5, 0.	"
3.7, -2., 1.5	"
4.8, -2.7, 0.	"
8	number of faces
1, 2, 4, 3, 0, 0	number of each corner in
3, 4, 6, 5, 0, 0	the face going in a
5, 6, 8, 7, 0, 0	clockwise direction
7, 8, 10, 9, 0, 0	"
9, 10, 12, 11, 0, 0	"
11, 12, 2, 1, 0, 0	"
1, 3, 5, 7, 9, 11	"
2, 4, 6, 8, 10, 12	"

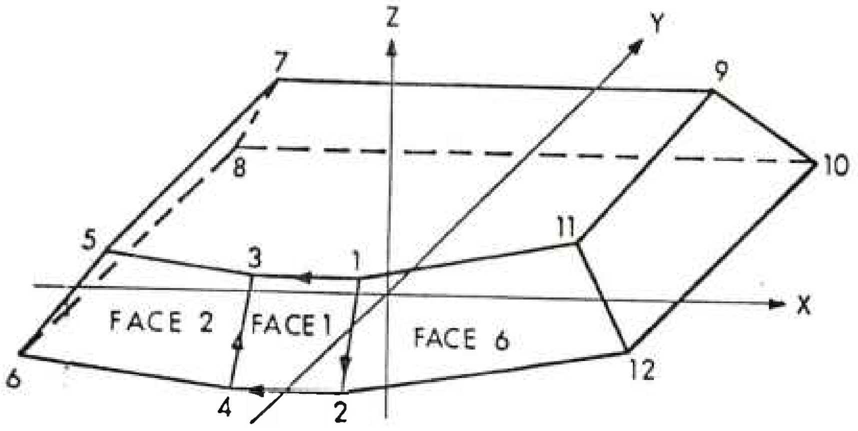

Figure 1. A Tank Turret

Level two on the other hand will produce a moderate amount of output data. With your input data being echoed you may do another check to make sure the data are all correct so that the solution that is printed is correct. If the target is not hit it simply will not print which faces were entered and exited. A sample level two output looks like this:

TABLE 2. SAMPLE OUTPUT FOR PRINT LEVEL TWO

```
                             x          y          z
Projectile velocity       3.000      4.000       .000
Projectile position      -2.000       .000      1.000
Positions of the 12 corners:
                           .900     -4.300      1.500
                           .900     -5.300       .000
                          -.900     -4.300      1.500
                          -.900     -5.300       .000
                         -3.700     -3.200      1.500
                         -4.800     -3.900       .000
                         -3.700      6.200      1.500
                         -4.800      7.500       .000
                          3.700      6.200      1.500
                          4.800      7.500       .000
                          3.700     -2.000      1.500
                          4.800     -2.700       .000
Corners of the  8 faces:
           1         2          4          3          0          0
           3         4          6          5          0          0
           5         6          8          7          0          0
           7         8         10          9          0          0
           9        10         12         11          0          0
          11        12          2          1          0          0
           1         3          5          7          9         11
           2         4          6          8         10         12
Projectile entered thru face 3
Projectile exited  thru face 4
```

Level three contains the maximum amount of data output. If doing any work that involves very specific data, you should use level three. It prints all of the same output as in level two plus the new projectile position after rotation, the new corner coordinates after rotation and translation, the number of corners in each face, the corner numbers with their vector coordinates, and the associated edge numbers, vectors, and cross products. Just like level two, level three will not print which faces were entered and exited if the target was not hit. A sample level three output would look like this:

TABLE 3. SAMPLE OUTPUT FOR PRINT LEVEL THREE

```
                              x         y         z
Projectile velocity       3.000     4.000      .000
Projectile position      -2.000      .000     1.000
Positions of the 12 corners:
                           .900    -4.300     1.500
                           .900    -5.300      .000
                          -.900    -4.300     1.500
                          -.900    -5.300      .000
                         -3.700    -3.200     1.500
                         -4.800    -3.900      .000
                         -3.700     6.200     1.500
                         -4.800     7.500      .000
                          3.700     6.200     1.500
                          4.800     7.500      .000
                          3.700    -2.000     1.500
                          4.800    -2.700      .000
Corners of the  8 faces:
         1         2         4         3         0         0
         3         4         6         5         0         0
         5         6         8         7         0         0
         7         8        10         9         0         0
         9        10        12        11         0         0
        11        12         2         1         0         0
         1         3         5         7         9        11
         2         4         6         8        10        12
New projectile pos:      -1.600    -1.200     1.000
New corner positions are:
                          4.900    -2.900      .500
                          5.500    -3.700    -1.000
                          3.460    -3.980      .500
                          4.060    -4.780    -1.000
                           .560    -4.780      .500
                           .100    -6.000    -1.000
                         -5.080     2.740      .500
                         -6.740     3.120    -1.000
                           .840     7.180      .500
                           .940     8.880    -1.000
                          5.760      .620      .500
                          7.060      .720    -1.000
Face  1 has 4 corners.
Corner vector  3 is:      3.460    -3.980      .500
Edge vector    1 is:      1.440     1.080      .000 cross product is:      .720
Corner vector  1 is:      4.900    -2.900      .500
Edge vector    2 is:       .600     -.800    -1.500 cross product is:     7.650
Corner vector  2 is:      5.500    -3.700    -1.000
Edge vector    4 is:     -1.440    -1.080      .000 cross product is:     1.440
Corner vector  4 is:      4.060    -4.780    -1.000
Edge vector    3 is:      -.600      .800     1.500 cross product is:    -5.490
```

TABLE 3 CONTD.

```
Face  2 has 4 corners.
Corner vector  5 is:      .560    -4.780      .500
Edge vector    3 is:     2.900      .800      .000 cross product is:     1.450
Corner vector  3 is:     3.460    -3.980      .500
Edge vector    4 is:      .600     -.800    -1.500 cross product is:     5.490
Corner vector  4 is:     4.060    -4.780    -1.000
Edge vector    6 is:    -3.960    -1.220      .000 cross product is:     3.960
Corner vector  6 is:      .100    -6.000    -1.000
Edge vector    5 is:      .460     1.220     1.500 cross product is:     -.610
Face  3 has 4 corners.
Corner vector  7 is:    -5.080     2.740      .500
Edge vector    5 is:     5.640    -7.520      .000 cross product is:     2.820
Corner vector  5 is:      .560    -4.780      .500
Edge vector    6 is:     -.460    -1.220    -1.500 cross product is:      .610
Corner vector  6 is:      .100    -6.000    -1.000
Edge vector    8 is:    -6.840     9.120      .000 cross product is:     6.840
Corner vector  8 is:    -6.740     3.120    -1.000
Edge vector    7 is:     1.660     -.380     1.500 cross product is:     8.450
Projectile entered thru face 3
Face  4 has 4 corners.
Corner vector  9 is:      .840     7.180      .500
Edge vector    7 is:    -5.920    -4.440      .000 cross product is:    -2.960
Corner vector  7 is:    -5.080     2.740      .500
Edge vector    8 is:    -1.660      .380    -1.500 cross product is:    -8.450
Corner vector  8 is:    -6.740     3.120    -1.000
Edge vector   10 is:     7.680     5.760      .000 cross product is:    -7.680
Corner vector 10 is:      .940     8.880    -1.000
Edge vector    9 is:     -.100    -1.700     1.500 cross product is:    -1.310
Projectile exited  thru face 4
Face  5 has 4 corners.
Corner vector 11 is:     5.760      .620      .500
Edge vector    9 is:    -4.920     6.560      .000 cross product is:    -2.460
Corner vector  9 is:      .840     7.180      .500
Edge vector   10 is:      .100     1.700    -1.500 cross product is:     1.310
Face  6 has 4 corners.
Corner vector  1 is:     4.900    -2.900      .500
Edge vector   11 is:      .860     3.520      .000 cross product is:      .430
Corner vector 11 is:     5.760      .620      .500
Edge vector   12 is:     1.300      .100    -1.500 cross product is:     9.290
Corner vector 12 is:     7.060      .720    -1.000
Edge vector    2 is:    -1.560    -4.420      .000 cross product is:     1.560
Corner vector  2 is:     5.500    -3.700    -1.000
Edge vector    1 is:     -.600      .800     1.500 cross product is:    -7.650
```

TABLE 3 CONTD.

```
Face  7 has 6 corners.
Corner vector 11 is:     5.760      .620      .500
Edge vector    1 is:     -.860    -3.520      .000 cross product is:     -.430
Corner vector  1 is:     4.900    -2.900      .500
Edge vector    3 is:    -1.440    -1.080      .000 cross product is:     -.720
Corner vector  3 is:     3.460    -3.980      .500
Edge vector    5 is:    -2.900     -.800      .000 cross product is:    -1.450
Corner vector  5 is:      .560    -4.780      .500
Edge vector    7 is:    -5.640     7.520      .000 cross product is:    -2.820
Corner vector  7 is:    -5.080     2.740      .500
Edge vector    9 is:     5.920     4.440      .000 cross product is:     2.960
Face  8 has 6 corners.
Corner vector 12 is:     7.060      .720    -1.000
Edge vector    2 is:    -1.560    -4.420      .000 cross product is:     1.560
Corner vector  2 is:     5.500    -3.700    -1.000
Edge vector    4 is:    -1.440    -1.080      .000 cross product is:     1.440
Corner vector  4 is:     4.060    -4.780    -1.000
Edge vector    6 is:    -3.960    -1.220      .000 cross product is:     3.960
Corner vector  6 is:      .100    -6.000    -1.000
Edge vector    8 is:    -6.840     9.120      .000 cross product is:     6.840
Corner vector  8 is:    -6.740     3.120    -1.000
Edge vector   10 is:     7.680     5.760      .000 cross product is:    -7.680
```

MATHEMATICS

This section describes the inner workings of the program. The program is divided into three parts related as shown in Figure 2. The MAIN routine is simply a driver program that reads inputs, echoes them as desired, and calls the two subsidiary routines. The XFORM subroutine rotates the target and translates it to a new coordinate system such that the projectile is traveling up the new y axis and passes through the origin. The HIT TGT function takes these new coordinates and examines each face to find if the projectile passes through that face.

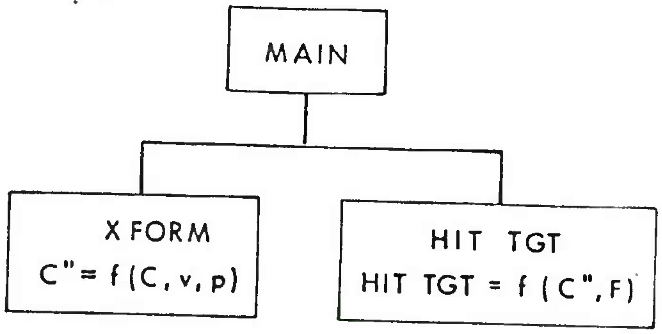

Figure 2. Program Heirarchy

Data Requirements

The coordinates are assumed to form a right-handed coordinate system and may be thought of as an x-axis positive toward the East, a y-axis positive toward the North, and a z-axis positive upward. Angles are measured from the y-axis clockwise.

The projectile is assumed to be in the vicinity of the target with a known position $\vec{p}$, and known velocity $\vec{v}$. The vertical or z component of the velocity vector is assumed to be zero. The target is assumed to be one or more polyhedra with convex faces, and is assumed to be described by its n corners in a matrix C of n rows and 3 columns. If a face is concave, it can be judiciously divided into several convex polygons when input is prepared. It is further assumed that each face has 3 to 6 corners and that the corners of the n faces are listed in a matrix F with n rows and 6 columns. The nth row contains a list of the m corners clockwise around the nth face as one looks from the outside of the polyhedron. If m is less than 6, the remaining elements of the row are zeroes.

Transformation of Target and Projectile Coordinates

The first step is to transform the coordinates of the target to a new coordinate system such that the projectile is traveling up the y-axis. The XFORM routine uses $\vec{p}$, $\vec{v}$, and C to produce a new matrix C'' which is the transformed matrix of corners, with the vector of each corner on a separate row. For the projectile and each corner vector, the rotation is equivalent and is as follows:

$$c_x' = c_x \cos\theta + c_y \sin\theta$$

$$c_y' = -c_x \sin\theta + c_y \cos\theta$$

$$c_z' = c_z$$

where $\theta = -atan(v_y, v_x)$, $\vec{c}$ is the original vector and c' is the vector after rotation.

Figure 3 illustrates the rotation and subsequent translation. The equations for the translation are:

$$c_x'' = c_x' - p_x'$$

$$c_z'' = c_z' - p_z'$$

$$c_y'' = c_y'.$$

In practice, the program uses the C matrix directly to produce a C'' matrix rather than using $\vec{c}$, $\vec{c}'$, and $\vec{c}''$ vectors for the individual rows.

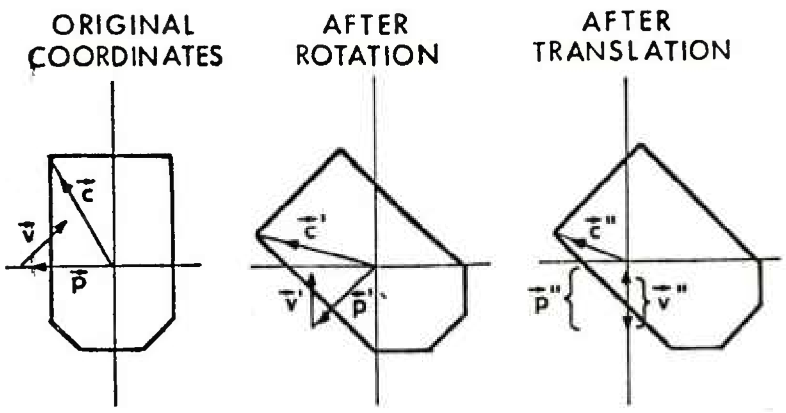

Figure 3. Transformation of Target Coordinates

Hit Determination

The HIT TGT function uses the C' matrix, and the F pointer matrix to find whether the target is hit. To do this, the program examines the silhouettes of the faces. If the sides of a silhouette encompass the origin of the coordinate system, the face is hit, but not otherwise. Figure 4 shows two silhouettes; the one on the left is hit and the one on the lower right is missed.

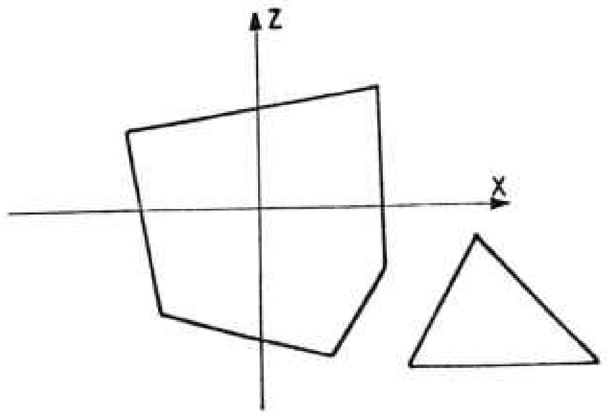

Figure 4. Target Faces Hit or Missed

To find whether a face is hit, the program examines the edges of its silhouette in turn. It uses the F pointer matrix to strip the two corners associated with an edge out of the C' matrix and place them in $\vec{c}$, and $\vec{c}'$ (not to be confused with vectors of the same name used in the XFORM subroutine.) Assume that $\vec{c}$, is a corner of the silhouette, $\vec{c}'$ is the corner immediately clockwise, and $\vec{e}$ is a vector representing the edge between these two corners. Then, as shown in Figure 5,

$$\vec{e} = \vec{c}' - \vec{c} .$$

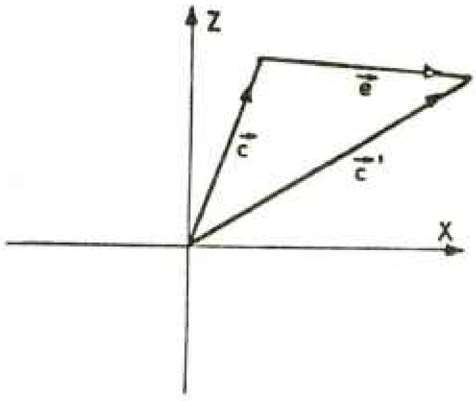

Figure 5. Edge and Corner Vectors

Now for $\vec{c}$, $\vec{e}$, the cross product will be a vector along the y axis. Notice that if the y component of the result is positive, the projectile can hit the silhouette because both the projectile (i.e. the origin) and the silhouette lie on the right of $\vec{e}$, that is, they are in the same half-plane. If the result is negative, however, the origin and the silhouette lie on opposite sides of the edge and no hit is possible. Mathematically, a hit is possible if and only if $r_y > 0$, where

$$r_y = c_z\, e_x - c_x\, e_z .$$

If all the cross products for the edges are positive in y, then the face is entered. Notice that a face on the far side of the target has its corners ordered clockwise when viewed from the outside of the target, but when viewed from the inside or behind, they are counter-clockwise. This allows us to find which faces are exited; they will be the faces whose cross products are all negative.

In summary, if a face has all positive cross products, it is entered; if a face has all negative cross products, it is exited; and if the cross products are a mixture of positive and negative, it is not hit. If we just want to know whether a target is entered, we examine each face until a negative cross product is found and examine faces until we find one that is hit. Notice also that we can determine whether an unclosed polyhedron is hit if we check both entries and exits.

Mechanization of the Algorithm

The correspondence between the notation discussed in this algorithm and that used in the actual program is shown in Table 2.

TABLE 4. CORRESPONDENCE BETWEEN ALGORITHM AND PROGRAM NOTATIONS

	NOTATION USED IN	
	Algorithm	Program
XFORM subroutine	C	corner
	C'	cornr2
	$\vec{p}$	p
	$\vec{v}$	v
HIT TGT function	$\vec{c}$	c
	$\vec{e}$	e
	y	y
	F	faces
	C'	cornr2

In actual practice, the y column of the C, and C' matrices is never used because for the silhouettes all values are zero.

SUMMARY

The program is a elegant algorithm for determining hits on irregular targets. It will find whether a hit occurs, and if desired, which faces are entered and exited. The target may be any open or closed polyhedron or a set of polyhedra. The elegance is due in part to the algorithm's careful organization and its ability to look at a single edge of a face and often determine that the projectile does not enter the face.

There are several modifications that could be made to this model. The ability to handle projectiles that are ascending or descending is a possible modification. The model could also be modified to find the angle at which the projectile strikes the faces. The structure makes it easy to take the two subsidiary routines and include them in larger combat models or to take the XFORM subroutine alone and use it to find silhouettes for other purposes. It can also be easily changed to run faster if only the faces entered are considered.

APPENDIX: PROGRAM LISTING

```
      parameter (NN=20)
c     PURPOSE: Find whether a projectile strikes a polyhedral target
      logical hit tgt, hit
      common /comtgt/ corner(NN,3), iface(NN,6), ncorns, nfaces
      common /projec/ v(3), p(3), lev
      real cornr2(NN,3)
1     format(26x,'x          y          z')
2     format(' Projectile velocity',3f10.3)
3     format(' Projectile position',3f10.3)
4     format(' Positions of the',i3,' corners:')
5     format(' Corners of the',i3,' faces:')
6     format(20x,3f10.3)
7     format(6i10)
c
c     Read inputs
          read *,lev
          read *,v
          read *,p
          read *,ncorns
          read *,((corner(i,j),j=1,3),i=1,ncorns)
          read *,nfaces
          read *,((iface(i,j),j=1,6),i=1,nfaces)
      IF (lev.gt.1) THEN
          print 1
          print 2,v
          print 3,p
          print 4, ncorns
          print 6,((corner(i,j),j=1,3),i=1,ncorns)
          print 5, nfaces
          print 7,((iface(i,j),j=1,6),i=1,nfaces)
      ENDIF
      call xform(cornr2)
      hit = hit tgt(cornr2)
      END
c
      LOGICAL FUNCTION HIT TGT (cornr2)
c     ---------------------------
c     Purpose: Find whether target is hit
      parameter (NN=20)
      logical hit fac
      common /comtgt/ corner(NN,3), iface(NN,6), ncorns, nfaces
      common /projec/ v(3), p(3), lev
      dimension c(3), e(3), cornr2(NN,3)
1     format(' Corner vector',i3,' is:',3f10.3)
2     format(' Edge vector  ',i3,' is:',3f10.3,' cross product is:',f10.3)
```

```
3     format(' Face',i3,' has',i2,' corners.')
c
      hit tgt=.false.
c     Check each face to see if it was hit
      DO 50 n=1,nfaces
          hit fac=.false.
c         Find j, the number of corners of face n
           j=0
           DO 20 k=1,6
               IF (iface (n,k) .gt. 0) j=j+1
20         CONTINUE
           if (lev.eq.3) print 3,n,j
          k=iface(n,j)
          nneg = 0
          npos = 0
c         Check each edge to see if the polygon and origin are in the
c         same half plane.
          DO 30 l=1,j
c          Load corner vector
               c(1)=cornr2(k,1)
               c(2)=cornr2(k,2)
               c(3)=cornr2(k,3)
               if (lev.eq.3) print 1,k,c
c          Load edge vector
               k=iface(n,l)
               e(1)=cornr2(k,1) - c(1)
               e(2)=cornr2(k,2) - c(2)
               e(3)=cornr2(k,3) - c(3)
           y=e(1)*c(3)-c(1)*e(3)
           if (lev.eq.3) print 2,k,e,y
           if (y .lt. 0) nneg = nneg + 1
           if (y .gt. 0) npos = npos + 1
           if ((nneg .ge. 1) .and. (npos .ge. 1)) GOTO 40
30        CONTINUE
40        CONTINUE
          if (npos.eq.j .or. nneg.eq.j) hit fac = .true.
          IF (lev.gt.1 .and. hit fac) THEN
           if (nneg.eq.j) print*,'Projectile exited  thru face',n
           if (npos.eq.j) print*,'Projectile entered thru face',n
          ENDIF
          hit tgt=hit tgt .or. hit fac
          if (lev .eq. 1 .and. hit tgt) GO TO 60
50    CONTINUE
60    CONTINUE
      if (lev.eq.1 .and. hit tgt) print*,'Yes, it was hit.'
      if (.NOT. hit tgt) print*,'No, it was not hit.'
      END
c
```

```
      SUBROUTINE XFORM(cornr2)
c     ------------------
c     PURPOSE: Rotate and translate target, proj to proj base coords
      parameter (NN=20)
      common /comtgt/ corner(NN,3), iface(NN,6), ncorns, nfaces
      common /projec/ v(3), p(3), lev
      dimension cornr2(NN,3)
1     format(' New projectile pos:',3f10.3)
2     format(' New corner positions are:')
3     format(20x,3f10.3)
c
c     Find sin, cos of rotation angle
          factor = 1./sqrt(v(1)**2 + v(2)**2)
          sina = -v(1)*factor
          cosa =  v(2)*factor
c     Rotate projectile position
          xp = p(1)*cosa + p(2)*sina
          yp =-p(1)*sina + p(2)*cosa
          if (lev.eq.3) print 1, xp, yp, p(3)
c     Rotate & translate coordinates of target corners
          DO 20 nc=1,ncorns
           cornr2(nc,1) = corner(nc,1)*cosa + corner(nc,2)*sina - xp
           cornr2(nc,2) = -corner(nc,1)*sina + corner(nc,2)*cosa
           cornr2(nc,3) = corner(nc,3) - p(3)
20        CONTINUE
          if (lev.eq.3) print 2
          if (lev.eq.3) print 3,((cornr2(i,j),j=1,3),i=1,ncorns)
      END
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
1	HQDA DAMA-ART-M Washington, DC 20310
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801-5001
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TSS Dover, NJ 07801-5001
1	Commander US Army Armament, Munitions & Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299-6000
1	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Aviation Research and Development Command ATTN: AMSAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120

No. of Copies	Organization
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications- Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmoutn, NJ 07703-5301
1	Commander US Army Tank Automotive Command ATTN: AMSTA-TSL Warren, MI 48090
1	Commander US Army Missile Command ATTN: AMSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-YDL Redstone Arsenal, AL 35898
2	Commander US Army Tank Automotive Command ATTN: AMSTA-ZEC, Mr. Hoeltzel (2 cys) Warren MI 48090
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missle Range, NM 88002

DISTRIBUTION LIST

No. of Copies	Organization
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, FL 32542-5000
1	President US Army Armor & Engineer Board ATTN: ATZK-AE Fort Knox, KY 40121-5000
1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Director DARCOM-MSA-ADEA ATTN: DRXTB-T, MAJ Martin Fort Lewis, WA 98433
1	Commander US Army Development & Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	AFWL/SUL Kirtland AFB, NM 87117
1	Commander US Naval Surface Weapon Ctr ATTN: G31 Dahlgren, VA 22448
1	Commander US Marine Corps Development and Education Command (MCDEC) ATTN: Plans & Studies Div Quantico, VA 22134

No. of Copies	Organization
	<u>Aberdeen Proving Ground</u>
	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, Mr. H. Cohen Mr. R. Mezan Mr. G. Comstock Ms. C. Harrington Ms. K. Tarquini Mr. C. Ehrig Mr. J. Meredith
	Cdr, USATECOM ATTN: AMSTE-TO-F
	Cdr, CRDC, AMCCOM ATTN: SMCCR-RSP-A SMCCR-MU SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number ____________________ Date of Report ____________

2. Date Report Received ____________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ____________________

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) ____________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate. ____________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)